UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/072,358	02/06/2002	Kenneth C. Duisenberg	10019681-1	4285

22879 7590 04/09/2008

HEWLETT PACKARD COMPANY
P O BOX 272400, 3404 E. HARMONY ROAD
INTELLECTUAL PROPERTY ADMINISTRATION
FORT COLLINS, CO 80527-2400

EXAMINER	
LEE, CHUN KUAN	
ART UNIT	PAPER NUMBER
2181	

NOTIFICATION DATE	DELIVERY MODE
04/09/2008	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

JERRY.SHORMA@HP.COM
mkraft@hp.com
ipa.mail@hp.com

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 10/072,358
Filing Date: February 06, 2002
Appellant(s): DUISENBERG, KENNETH C.

Hewlett-Packard Development Company, L.P.
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 01/28/2008 appealing from the Office action mailed 08/10/2007.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

Applicant's Admitted Prior Art, Applicant's Application in the Specification pages 1-5 and Figure 1.

US Patent 4,637,023 Lounsbury et al. 01-1987

US Patent 5,860,001 Cromer et al. 01-1999

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 1-2, 4-17 and 19-24 are rejected under 35 U.S.C. 103(a) as being unpatentable over Applicant's Admitted Prior Art (AAPA) in view of Lounsbury et al. (US Patent 4,637,023).

As per claims 1, 10 and 16, AAPA teaches a data processing flow control computer system and method comprising:

a processor (Specification, p. 1, ll. 10-11);

a ring of buffers (Drawings, Fig. 1 and Specification, p. 1, l. 17);

a computer readable memory coupled to said processor implementing a method of processing data (Specification, p. 1, ll. 16-34), comprising:

receiving an interrupt indicating data from a local area network (LAN) has been stored in one of a plurality of buffers and is ready for processing (Specification, p. 1, ll. 19-21);

if a software buffer index points to a first buffer containing processed data as the LAN software check and finds the processed data, the LAN software waits and the software buffer index is not advanced to the next buffer (Specification, p. 3, ll. 9-15);

if the software buffer index points to the first buffer containing unprocessed data (Specification, p. 1, ll. 21-22), as the LAN software check and finds the new data to process, the unprocessed data is processed by the LAN software and the software buffer index is advanced to the next buffer (Specification, p. 4, ll. 24-26);

sequentially processing the plurality of subsequent buffers if the LAN software continue to check and find unprocessed data in the subsequent buffers, then advancing the software buffer index after processing the unprocessed data, wherein the checking, finding and processing of unprocessed data is implemented until the LAN software check and finds processed data in the subsequent buffer (Specification, p. 4, ll. 24-30); and

when the software buffer index is synchronized to a hardware buffer index, after processing the unprocessed data, the software buffer index is reset to a next available buffer having processed data (Specification, p. 3, l. 33 to p. 4, l. 9).

AAPA does not teach the data processing flow control computer system and method comprising:

searching through the plurality of buffers containing data to determine whether there is a second buffer with unprocessed data, if the software buffer index points to the first buffer containing processed data; and

if there is said second buffer with unprocessed data, synchronizing said software buffer index to a hardware buffer index by resetting said software buffer index to a next available buffer having processed data following said second buffer, and otherwise stopping said searching when each buffer of said plurality of buffers has been searched and a buffer with unprocessed data is not found.

Lounsbury teaches a system and a method comprising:

transferring data between a peripheral and a host computer via a ring of buffers (col. 13, ll. 3-47): and

when an error occurs (e.g. existence of invalid data as there is unprocessed data in said second buffer when the software buffer index is pointing to the current first buffer contain processed data) while transferring the data from the ring of buffers to the host computer, the ring of buffers is sequentially searched for the correct data to continue transferring, and stop the searching if the correct data is not found after searching all the buffers in the ring of buffers (col. 13, ll. 18-41 and col. 18, ll. 41-52); therefore, if there is said second buffer with unprocessed data when the software buffer index is pointing to the current first buffer contain processed data (e.g. error associated with existence of invalid data in the ring buffer), resetting said software buffer index to a next available buffer having processed data following said second buffer by sequential searching to find and correct the error, and after the error is corrected by processing the unprocessed data in said second buffer, said software buffer index would be reset to the next available buffer having processed data following said second buffer.

It would have been obvious to one of ordinary skill in this art, at the time of invention was made to include Lounsbury's searching of the ring of buffers into AAPA's data processing flow control for the benefit of providing a robust method for transferring data between the peripheral and the host computer via the ring of buffers, by correcting the error having invalid data in the ring of buffers (e.g. having unprocessed data in said second buffer when the software buffer index is pointing to the current first buffer contain processed data) as data is transferred from the ring of buffers to the host computer are corrected (Lounsbury, Abstract; col. 13, ll. 18-41 and col. 18, ll. 41-52). The resulting combination of the references further teaches the data processing flow control computer system and method comprising:

when the error occurs as the software buffer index pointing to the first buffer containing processed, the ring of buffers is sequentially searched for a second buffer with the unprocessed data;

if the second buffer with the unprocessed data was found, the software buffer index would synchronize with the hardware buffer index, as the detected unprocessed data and any subsequent unprocessed data would be processed until the software buffer index reaches the buffer with processed data, therefore resetting the software buffer index to the next available buffer having processed data following the second buffer; and

stopping the searching of the second buffer with the unprocessed data if the unprocessed data is not found after searching all the buffers in the ring of buffers.

As per claims 2 and 17, AAPA and Lounsbury teach all the claimed limitations of claims 1 and 16 as discussed above, where AAPA further teaches the data processing flow control computer system and method wherein said synchronizing further comprises synchronizing hardware buffer index and said software buffer index in response to an interrupt indicating data has been stored in one of said plurality of buffers and is ready for processing (AAPA, Specification, p. 1, ll. 19-21).

As per claims 4, 13 and 19, AAPA and Lounsbury teach all the claimed limitations of claims 1, 10 and 16 as discussed above, where AAPA further teaches the data processing flow control computer system and method comprises determining if said first buffer contains processed data; and processing data in said first buffer if said data is unprocessed (AAPA, Specification, p. 1, ll. 21-22), as the LAN software check the buffer index and find the new data to process.

As per claims 5 and 20, AAPA and Lounsbury teach all the claimed limitations of claims 1 and 16 as discussed above, where AAPA further teaches the data processing flow control computer system and method wherein said synchronizing further comprises wrapping around to a start buffer (AAPA, Drawings, Buffer 0 of Fig. 1) after searching the end buffer in said plurality of buffers when sequentially searching through said plurality of buffers, said plurality of buffers sequentially beginning with the start buffer (AAPA, Drawings, Buffer 0 of Fig. 1) and ending with an end buffer (AAPA, Drawings, Buffer N of Fig. 1) (AAPA, Specification, p. 1, ll. 23-26 and p. 4, ll. 20-33).

As per claims 6, 14 and 21, AAPA and Lounsbury teach all the claimed limitations of claims 1, 10 and 16 as discussed above, where both further teach the data processing flow control computer system and method further comprises:

when said software buffer index points to said first buffer containing processed data, using a value of said software buffer index corresponding to said first buffer as a reference value (Lounsbury, col. 18, ll. 41-52), wherein it would have been obvious to utilize the value of the software buffer index as the reference point in order to properly determine if all the buffers in the ring of buffers have been searched;

incrementing said software buffer index as each buffer of said plurality of buffers is searched, wherein when said software buffer index reaches one end of a range of possible values it is reset to the other end of said range (AAPA, Drawings, Fig. 1; Specification, p. 1, ll. 23-26 and p. 4, ll. 27-33); and

stopping said searching when said software buffer index reaches said reference value without finding a buffer in said plurality of buffers with unprocessed data (Lounsbury, col. 18, ll. 41-52).

As per claims 7 and 22, AAPA and Lounsbury teach all the claimed limitations of claims 1 and 16 as discussed above, where AAPA further teaches the data processing flow control computer system and method further comprises wherein each of said plurality of buffers is a local area network (LAN) buffer for storing LAN packets of data (AAPA, Specification, p. 1, ll. 33-34).

As per claims 8 and 23, AAPA and Lounsbury teach all the claimed limitations of claims 7 and 22 as discussed above, where AAPA further teaches the data processing flow control computer system and method further comprises wherein said software buffer index is a LAN software buffer index, and said hardware buffer index is a LAN hardware buffer index (AAPA, Specification, p. 1, ll. 28-29).

As per claim 9, 15 and 24, AAPA and Lounsbury teach all the claimed limitations of claims 1, 10 and 16 as discussed above, where AAPA further teaches the data processing flow control computer system and method further comprises processing said unprocessed data in said second buffer (AAPA, Specification, p. 1, ll. 21-22 and p. 4, ll. 20-33), as the LAN software check the buffer index and find the new data to process.

As per claim 11, AAPA and Lounsbury teach all the claimed limitations of claim 10 as discussed above, where AAPA further teaches the data processing flow control computer system and method further comprises wherein said data from said LAN is a LAN packet (AAPA, Specification, p. 1, ll. 33-34).

As per claim 12, AAPA and Lounsbury teach all the claimed limitations of claim 10 as discussed above, where AAPA further teaches the data processing flow control computer system and method further comprises wherein a LAN driver performs said receiving, said searching and said synchronizing (AAPA, Specification, p. 1, ll. 12-14).

Claims 3 and 18 are rejected under 35 U.S.C. 103(a) as being unpatentable over Applicant's Admitted Prior Art (AAPA) in view of Lounsbury et al. (US Patent 4,637,023) as applied to claims 1 and 16, and further in view of Cromer et al. (US Patent 5,860,001).

AAPA and Lounsbury teach all the claimed limitations of claims 1 and 16 as discussed above, where AAPA further teaches the data processing flow control computer system and method comprising synchronizing the hardware buffer index and the software buffer index in response to a first interrupt indicating data has been stored in one of said plurality of buffers and is ready for processing when the software buffer index points to the first buffer containing processed data (AAPA, Specification, p. 1, ll. 19-21).

AAPA and Lounsbury do not teach the data processing flow control computer system and method wherein said synchronizing further comprises ignoring the first interrupt indicating data has been stored in one of said plurality of buffers and synchronizing said hardware buffer index and said software buffer index in response to a second interrupt indicating data has been stored in one of said plurality of buffers.

Cromer teaches the computer system and method comprising wherein there are at least two boot sequences and wherein the boot sequence after the computer is turned on comprising of loading BIOS, the operating system and the particular application defined by initialization control information which causes an initial program load (IPL) (col. 1, l. 28 to col. 2, l. 64), therefore upon turning on the computer system,

said computer system requires a period of time before reaching a state of stability wherein the properly initialization has been completed (Fig. 10-11).

It would have been obvious to one of ordinary skill in this art, at the time of invention was made to include Cromer's plurality of boot sequences into AAPA and Lounsbury's data processing flow control for the benefit of providing the easier method for managing the computer system (Cromer, col. 1, l. 28 to col. 2, l. 64 and col. 11, ll. 14-23). The resulting combination of the references further teaches the data processing flow control computer system and method wherein said synchronizing further comprises ignoring the first interrupt because the computer system requires the period of time before reaching the state of stability; and synchronizing the hardware buffer index and the software buffer index in response to a second interrupt indicating data has been stored in one of said plurality of buffers and is ready for processing when the software buffer index points to said first buffer containing processed data for a second time, as the state of stability is reached.

(10) Response to Argument

I. claims 1-2, 4-17 and 19-24

Issue I

It appears the appellant is arguing that Lounsbury is nonanalogous art, because appellant argued that checking for valid data or invalid data is substantially different from checking form processed or unprocessed data.

Examiner's response to Issue

The examiner respectfully disagree, as Lounsbury is in the field of applicant's endeavor, wherein Lounsbury's teaching is associated with transferring of data in a computer system via a ring of buffers (e.g. ring buffer) (Lounsbury, col. 1, ll. 12-15 and col. 13, ll. 3-34); additionally, Lounsbury's teaching is also reasonably pertinent to the particular problem with which the applicant was concerned, which is sequential searching and checking for an error (e.g. invalid data) in the ring of buffers (Lounsbury, col. 13, ll. 18-41 and col. 18, ll. 41-52).

Additionally, examiner is relying on Lounsbury's valid data for the teaching/suggesting of appellant's processed data and Lounsbury's invalid data for the teaching/suggesting of appellant's unprocessed data; wherein the examiner is applying Lounsbury's teaching that when the error occurs (e.g. existence of invalid data as there is unprocessed data in the second buffer on the ring of buffer) to implement the sequential search to find and correct error (e.g. error is corrected as the second buffer with unprocessed data is found and processed).

Issue II

The appellant argued that the combination of AAPA and Lounsbury do not teach/suggest every claimed limitation, because nothing in Lounsbury teaches a buffer index.

Examiner's response to Issue

The examiner respectfully disagrees, as in accordance to examiner's understand of the appellant's invention, the buffer index have a equivalent function to a pointer pointing to

the corresponding location within the ring buffer; wherein the use of the pointer in association with the ring buffer is well known to one skilled in the art. Additionally, the buffer index is also taught by AAPA.

Issue III

Appellant argued that the combination of references do not teach the claimed limitation of "if there is said second buffer with unprocessed data, resetting said software buffer index to a next available buffer having processed data following said second buffer;" additionally, the combination of the reference would not teach the above claimed limitation, but rather teach a system which can/would skip over unprocessed data in a buffer, rather then sequentially processing unprocessed data as it is accessed.

Examiner's response to Issue

The examiner respectfully disagrees, as the combination of references would teach/suggest the above limitation because the examiner is not relying on Lounsbury's teaching for skip over unprocessed data in a buffer. To further explain how the examiner relied on the references, in accordance to the preceding final office action, the reference are relied on as following:

AAPA teaches "resetting said software buffer index to a next available buffer having processed data" following a current buffer, wherein an interrupt is received and the software buffer index is pointing to said current buffer containing unprocessed data, the unprocessed data in said current buffer would then be processed following by resetting the

software buffer index to the next available buffer having processed data following said current buffer (Specification, page. 1, ll. 19-22 and page 4, ll. 24-31).

Lounsbury teaches "if there is said second buffer with unprocessed data", as Lounsbury teaches if there is an error associated with the ring buffer (e.g. error of having the second buffer with unprocessed data in the ring buffer when the software buffer index is pointing to the current first buffer contain processed data) while data is transferred via the ring buffer, a sequential search would be implemented to find and correct the error (e.g. sequentially searching to find the ring buffer for the error associated with said second buffer with unprocessed data and (correct the error by) processing the unprocessed data in said second buffer) (col. 13, ll. 3-47 and col. 18, ll. 41-52).

The resulting combination of AAPA and Lounsbury teaches/suggests if there is said second buffer with unprocessed data (e.g. an error in the ring buffer), resetting said software buffer index to a next available buffer having processed data following said second buffer, as the sequential searching to find and correct error is implemented, and after the error is corrected by processing the unprocessed data in said second buffer, said software buffer index would be reset to the next available buffer having processed data following said second buffer.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

Chun-Kuan Lee
Patent Examiner
Art Unit 2181

Conferees:

/Alford W. Kindred/

Supervisory Patent Examiner, Art Unit 2163

Alford W. Kindred
Supervisory Patent Examiner
Art Unit 2181

/Manorama Padmanabhan/
Manorama Padmanabhan
Workgroup QAS 2180